TOP 50 QUESTIONS

in hardik agnihotri

Q 1. What is React?

Ans: React is an open-source JavaScript library for building user interfaces or UI components, developed by Facebook.

Q 2. Explain JSX

Ans: JSX (JavaScript XML) is a syntax extension for JavaScript recommended by React for describing what the UI should look like.

Q 3. What is the virtual DOM?

Ans: The virtual DOM is a lightweight copy of the actual DOM in memory. React uses it to improve performance by updating only the changed parts of the actual DOM.

Q 4. What is the significance of keys in React?

Ans: Keys are used to uniquely identify and differentiate between components in React. They help React identify which items have changed, added, or removed.

Q 5. What are state and props in React?

Ans: State is an internal data store that belongs to a specific component, and it can be changed over time. Props are properties passed to a component from its parent, and they are immutable.

Q 6. What is the difference between state and props?

Ans: State is internal to a component and can be changed over time, while props are external and passed to a component.

Q 7. Explain the concept of lifting state up.

Ans: Lifting state up is a pattern where the state of a child component is moved to its parent component, allowing multiple child components to share the same state.

Q 8. What is the purpose of setState in React?

Ans: setState is used to update the state of a component and trigger a re-render.

Q 9. What is React Router?

Ans: React Router is a library that enables navigation among views in a React application, allowing for the development of single-page applications.

Q 10. Explain the useEffect hook.

Ans: The useEffect hook in React is used for side effects in functional components, such as data fetching, subscriptions, or manually changing the DOM.

Q 11. What are controlled components in React?

Ans: Controlled components are components where the form data is controlled by React state. The input elements receive their current value from the state and have their value updated through a callback function.

Q 12. What is Redux, and why is it used?

Ans: Redux is a state management library for JavaScript applications, commonly used with React. It helps manage the state of an application in a predictable way.

Q 13. Explain the concept of higher-order components (HOC).

Ans: Higher-order components are functions that take a component and return a new component with additional features or props.

Q 14. What is the purpose of the useReducer hook?

Ans: The useReducer hook is used for managing complex state logic in React applications. It is an alternative to useState when state transitions are more complex.

Q 15. What is the significance of the key attribute in React lists?

Ans: The key attribute is used to uniquely identify elements in a list. It helps React efficiently update the DOM when the list changes.

Q 16. What is the difference between class components and functional components?

Ans: class components use ES6 classes and have additional features like state and lifecycle methods, while functional components are simpler and are often used with hooks.

Q 17. Explain the concept of refs in React.

Ans: Refs are used to access the DOM directly or to reference a React element. They provide a way to interact with the underlying DOM nodes in React.

Q 18. What are React hooks?

Ans: React hooks are functions that allow functional components to use state, lifecycle methods, and other React features.

Q 19. Explain the purpose of the useContext hook.

Ans: The useContext hook is used to access the value of a React context within a functional component.

Q 20. What is the significance of the dangerouslySetInnerHTML property in React?

Ans: dangerouslySetInnerHTML is used to inject HTML directly into a component, but it should be used with caution to avoid cross-site scripting (XSS) vulnerabilities.

Q 21. What is the purpose of the componentDidMount lifecycle method?

Ans: componentDidMount is invoked immediately after a component is mounted, making it suitable for initial AJAX requests or setting up subscriptions.

Q 22. What is the React developer tool?

Ans: The React Developer Tools is a browser extension that allows developers to inspect and debug React component hierarchies in the Chrome and Firefox browsers.

Q 23. Explain the concept of context in React.

Ans: Context provides a way to pass data through the component tree without having to pass props manually at every level. It is often used to share values like themes or authentication status.

Q 24. What are the advantages of using React?

Ans: React offers a virtual DOM for improved performance, a component-based architecture for modular development, and a strong community support, among other advantages.

Q 25. How does React handle prop drilling, and how can it be avoided?

Ans: Prop drilling occurs when props are passed down through multiple levels of components. It can be avoided by using context or state management libraries like Redux.

Q 26. What is the purpose of the shouldComponentUpdate method?

Ans: shouldComponentUpdate is a lifecycle method that determines if a component should re-render. Developers can use it to optimize performance by preventing unnecessary renders.

Q 27. Explain the significance of React Fragments.

Ans: React Fragments allow developers to group multiple elements without adding an extra node to the DOM, helping to keep the structure clean.

Q 28. What is the significance of the key prop in React Router?

Ans: The key prop in React Router is used to force the remounting of a component when the key changes, ensuring that the component is fully reinitialized.

Q 29. What is the purpose of the forwardRef function in React?

Ans: forwardRef is used to forward refs through components, allowing parent components to interact with the child's DOM node.

Q 30. Explain the concept of error boundaries in React.

Ans: Error boundaries are components that catch JavaScript errors anywhere in their child component tree and log those errors, display a fallback UI, or take other actions.

Q 31. What is the significance of the memo function in React?

Ans: memo is a higher-order component that memoizes the rendering of a functional component, preventing unnecessary re-renders if the props have not changed.

Q 32. How does React handle forms?

Ans: React handles forms by using controlled components, where form data is controlled by the React state.

Q 33. Explain the purpose of the useMemo hook.

Ans: The useMemo hook is used to memoize the result of a function, preventing unnecessary calculations and improving performance.

Q 34. What is the significance of the useCallback hook?

Ans: useCallback is used to memoize callback functions, preventing them from being recreated on every render.

Q 35. What are React portals?

Ans: React portals provide a way to render children into a DOM node that exists outside the parent component's hierarchy.

Q 36. Explain the concept of suspense in React.

Ans: Suspense is a feature in React that allows components to "wait" for something before rendering, such as data fetching or code splitting.

Q 37. What is the purpose of the useEffect cleanup function?

Ans: The cleanup function in useEffect is used to perform cleanup tasks, such as unsubscribing from subscriptions or clearing intervals, when a component is unmounted.

Q 38. How does React handle routing?

Ans: React can handle routing using the React Router library, which provides a way to navigate between different views or pages in a React application.

Q 39. What is the purpose of the useLayoutEffect hook?

Ans: useLayoutEffect is similar to useEffect, but it fires synchronously after all DOM mutations. It is often used for measuring and synchronizing layout.

Q 40. Explain the concept of lazy loading in React.

Ans: Lazy loading is a technique where components or modules are loaded only when they are actually needed, improving initial load times.

Q 41. What is the significance of the React.memo function?

Ans: React.memo is a higher-order component that memoizes the rendering of a functional component, preventing unnecessary re-renders if the props have not changed.

Q 42. How does React handle code splitting?

Ans: React supports code splitting, allowing developers to split their code into smaller chunks that are loaded on demand, improving performance by reducing the initial bundle size.

Q 43. What is the purpose of the useImperativeHandle hook?

Ans: useImperativeHandle is used to customize the instance value that is exposed when using React.forwardRef.

Q 44. Explain the concept of the useDebugValue hook.

Ans: useDebugValue is used to display a label for custom hooks in React DevTools.

Q 45. What is the purpose of the useState hook?

Ans: The **useState** hook is used to add state to functional components in React.

Q 46. Explain the significance of the SuspenseList component in React.

Ans: SuspenseList is a component that allows developers to coordinate the loading of multiple components in a way that provides a better user experience.

Q 47. What is the significance of the react-scripts package in a React application?

Ans: react-scripts is a set of scripts and configurations used by Create React App to set up and manage a React project without the need for complex configuration.

Q 48. Explain the concept of error boundaries in React.

Ans: Error boundaries are components that catch JavaScript errors anywhere in their child component tree and log those errors, display a fallback UI, or take other actions.

Q 49. What is the purpose of the useReducer hook?

Ans: useReducer is a hook in React used for state management in functional components. It is particularly useful when the state logic is complex and involves multiple sub-values or when the next state depends on the previous state.

Q 50. How does React handle forms?

Ans: React handles forms using controlled components, where the form elements are controlled by React state. This allows React to be the single source of truth for the form data.

Did you find it Useful?

Leave a comment!

Like

Comment

Repost